

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-2. (canceled)

3. (currently amended) A method of rollbacking ~~at least one~~ a table of an active database to a point-in-time, the table having ~~at least one~~ a before image table and ~~at least one~~ an after image table, the method comprising:

retrieving a retention time for ~~each of the~~ the at least one table;

determining that ~~none of the retention times~~ time for the ~~at least one~~ table is not greater than the point-in-time;

~~locking all tables to be rollbacked~~ the table and disabling ~~all related associated referential~~ integrity constraints;

~~deleting rows from each of the~~ the after image table having a timestamp greater than the point-in-time; and

~~inserting into each of the~~ the after image table, rows from the ~~associated~~ before image table having a timestamp less than or equal to the point-in-time and having a changing timestamp greater than the point-in-time; and

~~enabling all related~~ the associated referential integrity constraints,

wherein the table is rollbacked to the point-in-time.

4. (currently amended) A method of rollbacking ~~at least one~~ a table of an active database to a point-in-time, the table having ~~at least one~~ a before image table and ~~at least one~~ an after image table, the method comprising:

retrieving a retention time for ~~each of the~~ the at least one table;

determining that ~~none of the retention times~~ time for ~~each of the~~ the at least one table is not greater than the point-in-time;

retrieving a transaction id set ~~associated with~~ comprising ~~[[a]]~~ any transaction id that begins before or at the point-in-time and ends after the point-in-time;

~~locking all tables to be rollbacked~~ the table and disabling all-related associated referential integrity constraints;

~~deleting rows from each of the at least one~~ the after image table having a timestamp greater than the point-in-time or having a transaction id that is a proper subset of the transaction id set;

~~inserting into each of the at least one~~ the after image table, rows from the ~~associated~~ before image table having a timestamp less than or equal to the point-in-time and having a transaction id that is not a proper subset of the transaction id set, and having a changing transaction id that is a proper subset of the transaction id set or having a changing timestamp that is greater than the point-in-time; and

~~enabling all-related~~ the associated referential integrity constraints,
wherein the table is rollbacked to the point-in-time.

5. (currently amended) A method of rollbacking ~~selected rows~~ a row in a table of an active database to a point-in-time, the database table having at least ~~one~~ a before image table and ~~at least one~~ an after image table, the method comprising:

retrieving a retention time of the table;

determining that the point-in-time is greater than or equal to the retention time;

~~locking the table to be rollbacked and disabling all-related~~ associated referential integrity constraints;

~~deleting selected rows~~ the row from the after image table having a timestamp greater than the point-in-time;

inserting into the after image table, ~~selected rows~~ a before image of the row from the before image table having a timestamp less than or equal to the point-in-time and having a changing timestamp that is greater than the point-in-time; and

enabling ~~all-related~~ the associated integrity constraints,
wherein the row is rolled back to the point-in-time.

6. (currently amended) A method of rolling back ~~selected rows~~ a row in a table of an active database to a point-in-time, the database table having at least ~~one~~ a before image table and ~~at least one~~ an after image table, the method comprising:

retrieving a transaction id set ~~associated with~~ comprising ~~[[a]]~~ any transaction id that begins before or at the point-in-time and ends after the point-in-time;

retrieving a retention time of the table;

determining that the point-in-time time is greater than or equal to the retention time;

locking the table to be rolled back and disabling ~~all-related~~ associated referential integrity constraints;

deleting ~~selected rows~~ the row from the after image table having a timestamp greater than the point-in-time or having a transaction id that is a member of the transaction id set;

inserting into the after image table, ~~selected rows~~ a before image of the row from the before image table having a timestamp that is less than or equal to the point-in-time and having a transaction id that is not a proper subset of the transaction id set, and having a changing transaction id that is a proper subset of the transaction id set or having a changing timestamp that is greater than the point-in-time; and

enabling ~~all-related~~ the associated referential integrity constraints,
wherein the row is rolled back to the point-in-time.

7-9. (canceled)

10. (currently amended) A method of rolling back a transaction id set in a ~~group~~ plurality of tables of an active database, each of the plurality of tables ~~database~~ having ~~at least one~~ a before image table and ~~at least one~~ an after image table, the method comprising:

retrieving an earliest starting time of the transaction id set;

retrieving a latest retention time for all each of the plurality of tables;
determining that the earliest starting time is greater than the latest retention time;
locking all each of the plurality of tables to be rolled back and disabling all ~~related~~ associated referential integrity constraints;
rollbacking the transaction id set in each of the ~~group~~ plurality of tables;
and
enabling all ~~related~~ the associated referential integrity constraints.

11. (currently amended) The method according to claim 10, further comprising:

deleting rows from each of the after image table tables having a an after image table transaction id that is a member of the transaction id set;

selecting earliest before image rows from each of the before image table tables having a before image table transaction id that is a proper subset of the transaction id set, or having a changing transaction id that is a proper subset of the transaction id set, and having an earliest timestamp;

deleting any after image of a row from each of the after image table tables when [[a]] the after image table transaction id of the row is a proper subset of the transaction id set and a changing transaction id of the row is not equal to a current transaction id; and

deleting any after image of a row from each of the after image table tables and inserting [[a]] the earliest before image from the row into each of the after image table tables when ~~the~~ a transaction id of the row is not a proper subset of the transaction id set and the changing transaction id is a proper subset of the transaction id set.

12. (currently amended) The method according to claim 10, further comprising:

deleting rows from each of the after image table tables having [[a]] an after image table transaction id that is a member of the transaction id set;

selecting before image rows and any descendants thereof from each of the before image table tables having a before image table transaction id that is a

proper subset of the transaction id set or having a changing transaction id that is a proper subset of the transaction id set;

grouping the selected before image rows into families;

removing any of the families having an earliest transaction id that is a proper subset of the transaction id set;

inserting an earliest returned before image in a family into each of the after image table tables when the last changing transaction id of the family is equal to a current transaction id; and

inserting an earliest returned before image in a family into each of the after image table tables when the last changing transaction id of the family is a proper subset of the transaction id set and if no after image of the family exists.

13. (canceled)

14. (currently amended) A system ~~for use~~ implemented with an active database ~~for performing at least one of online selectively rollbacking at least one application table or selected application data in at least one table, creating online point-in-time views of application tables, providing online history images associated with the database, reconstructing equivalent SQL statements of a committed transaction, reconstructing equivalent SQL statements of a user session, and providing a selective audit trail report on demand,~~ the system comprising:

a plurality of user application after image tables, wherein each of the tables has one before image table to store before images, and one after image view;

a first table ~~comprising~~ including a table name field, a retention time field, and an export timestamp field;

an image manager that creates image views; and

a rollback manager that manages the first table,

wherein the rollback manager is configured to rollback data in database tables and reconstruct SQL statements of committed transactions.

15. (original) The system according to claim 14, further comprising a second table including fields associated with a user session.

16. (original) The system according to claim 14, further comprising a third table including fields associated with a transaction id, starting timestamp, and ending timestamp.

17. (original) The system according to claim 14, the image manager further comprising:

an inserting trigger that sets parameters for an inserted row in the after image tables;

an updating trigger that inserts an original row into a before image table and sets parameters associated with the changed row in the after image tables and the before image tables; and

a deleting trigger that inserts the original row into the before image table and sets parameters associated with the changed row in the before image tables.

18. (original) The system according to claim 17, further comprising:
a transaction trigger that records each transaction id.

19. (currently amended) The system according to claim 17, further comprising:

a login/logout login trigger that records a time when a user enters the database and a logout trigger that records a time when a user ~~or~~ exits the database.

20. (currently amended) The system according to claim 14, the after image tables further comprising including at least one of a timestamp field, user id field, and transaction id field.

21. (currently amended) The system according to claim 14, ~~wherein the rollback manager performs at least one of rollbacking at least one table to a point-in-time, rollbacking selected rows in a single table to a point-in-time, rollbacking a transaction id set in a group of tables, rollbacking all transactions made in a user session, reconstructing equivalent SQL statements of a transaction id set in a group of tables, reconstructing equivalent SQL statements of a user session in a group of tables, and providing a selective audit trail report on-demand~~ wherein the data includes at least one of rows in the database tables, a transaction set, and a transaction made by a user.

22-30. (canceled)

31. (currently amended) A method of rollbacking transactions made in a user session in a group plurality of tables of an active database, each of the plurality of tables database having at least one a before image table and at least one an after image table, the method comprising:

retrieving each of a starting time, an ending time, and a unique session id of the user session;

retrieving a latest retention time for all each of the plurality of tables;

determining that the starting time is greater than the latest retention time of each of the plurality of tables;

locking all each of the plurality of tables to be rollbacked and disabling all related associated integrity constraints;

rollbacking the transactions made in the user session in the group plurality of tables; and

enabling all-related the associated integrity constraints.

32. (currently amended) The method according to claim 31, wherein the unique session id comprises ~~a unique session id~~ a unique identifier of the session provided by a database vendor.

33. (currently amended) The method according to claim 31, wherein the unique session id comprises a session id and the a starting timestamp of the session.

34. (currently amended) The method according to claim 31, further comprising:

deleting rows from each of the after image table tables having a unique session id that is equal to the user session's unique session id;

selecting before image rows, having an earliest timestamp, from each of the before image table tables having a unique session id that is equal to the user session's unique session id of the user session, or having a changing unique session id that is equal to the user session's unique session id;

deleting any after image of a row from each of the after image table tables [[if]] when a unique session id of the row is equal to the user session's unique

session id and a changing unique session id of the row is not equal to a current session's unique session id; and

deleting any after image of a row from each of the after image table tables and inserting a before image from the row into each of the after image table tables if the unique session id of the row is not equal to the user session's unique session id set and the changing unique session id is equal to the user session's unique session id.

35. (canceled)

36. (new) The method according to claim 31, further comprising:

deleting rows from each of the after image tables whose unique session id is equal to the unique session id of the user session;

selecting before image rows and any descendants thereof from each of the before image tables whose unique session id is equal to the unique session id of the user session or having a changing unique session id that is equal to the unique session id of the user session;

grouping the selected rows into families;

removing any families having an earliest unique session id that is equal to the unique session id of the user session;

inserting an earliest returned before image in a family into each of the after image tables when a last changing unique session id of the family is equal to a unique session id of a current session; and

inserting an earliest returned before image in a family into each of the after image tables when the last changing unique session id of the family is equal to a unique session id of the user session and if no after image of the family exists.

37. (new) The system according to claim 14, wherein each of the before image tables includes a changing timestamp field, a changing user id field, a changing transaction id field, an op code field, and fields of an associated after image table.